



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출원번호 : 10-2003-0021114  
Application Number

출원년월일 : 2003년 04월 03일  
Date of Application APR 03, 2003

출원인 : 삼성전자주식회사  
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



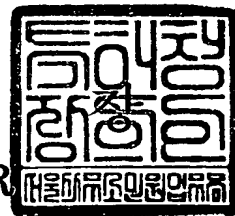
2003 년 10 월 10 일

특

허

청

COMMISSIONER





1020030021114

출력 일자: 2003/10/17

【서지사항】

【서류명】	특허출원서		
【권리구분】	특허		
【수신처】	특허청장		
【참조번호】	0001		
【제출일자】	2003.04.03		
【발명의 명칭】	에러정정기능을 가진 플래쉬메모리장치		
【발명의 영문명칭】	FLASH MEMORY WITH ERROR CORRECTION FOR PAGE COPY		
【출원인】			
【명칭】	삼성전자 주식회사		
【출원인코드】	1-1998-104271-3		
【대리인】			
【성명】	임창현		
【대리인코드】	9-1998-000386-5		
【포괄위임등록번호】	1999-007368-2		
【대리인】			
【성명】	권혁수		
【대리인코드】	9-1999-000370-4		
【포괄위임등록번호】	1999-056971-6		
【발명자】			
【성명의 국문표기】	이진엽		
【성명의 영문표기】	LEE, JIN YUB		
【주민등록번호】	700212-1011313		
【우편번호】	137-030		
【주소】	서울특별시 서초구 잠원동 신반포 4차APT 207동 904호		
【국적】	KR		
【심사청구】	청구		
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 임창현 (인) 대리인 권혁수 (인)		
【수수료】			
【기본출원료】	20	면	29,000 원
【가산출원료】	12	면	12,000 원



1020030021114

출력 일자: 2003/10/17

【우선권주장료】	0	건	0	원
【심사청구료】	19	항	717,000	원
【합계】	758,000			원
【첨부서류】	1. 요약서·명세서(도면)_1통			

**【요약서】****【요약】**

본 발명은 낸드형 플래쉬메모리에서의 카피백에 관한 것이다. 고유의 비트에러가 전사되는 것을 방지하기 위하여, 본 발명에 따른 낸드형 플래쉬메모리는, 페이지버퍼에 특정 페이지의 저장된 상기 원데이타의 비트에러를 정정하는 회로, 상기 원데이타를 상기 회로로 제공하고 상기 회로에 의해 정정된 보정데이타를 상기 페이지버퍼로 제공하는 수단, 그리고 상기 원데이타를 상기 페이지버퍼로 복사하고 상기 페이지버퍼로부터 상기 보정데이타를 다른 페이지에 저장하는 수단을 포함한다.

**【대표도】**

도 4

**【색인어】**

카피백, 페이지버퍼, 에러정정

【명세서】

【발명의 명칭】

에러정정기능을 가진 플래쉬메모리장치{FLASH MEMORY WITH ERROR CORRECTION FOR PAGE COPY}

【도면의 간단한 설명】

도 1은 낸드형 플래쉬메모리에서 종래의 카피백 동작을 보여주는 도면이다.

도 2는 낸드형 플래쉬메모리에서 본 발명에 따른 카피백시의 에러정정 과정을 개략적으로 보여주는 도면이다.

도 3은 본 발명에 따른 에러정정회로의 구성도이다.

도 4는 페이지버퍼와 에러정정회로사이의 데이터전송을 위한 회로 구성을 보여주는 회로도이다.

도 5는 페이지버퍼와 에러정정회로 사이의 데이터 전송과정을 개략적으로 보여주는 타이밍도이다.

도 6은 본 발명에 따른 컬럼패리티 및 라인패리티 발생 방식을 보여주는 도면이다.

도 7은 도 6에 보인 컬럼패리티를 발생하는 회로도이다.

도 8은 도 6에 보인 라인패리티를 발생하는 회로도이다.

도 9는 본 발명에 따라 페이지버퍼로부터 에러정정회로로의 데이터전송과정을 상세하게 보여주는 타이밍도이다.

도 10은 본 발명에 따른 에러정정과정을 보여주는 순서도이다.

본 발명에 따른 도면들에서 실질적으로 동일한 구성과 기능을 가진 구성요소들에 대하여는 동일한 참조부호를 사용한다.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

- <12> 본 발명은 플래쉬메모리에 관한 것으로서, 특히 카피백 동작 이후에 비트에러를 감지하고 정정(error check, detection and correction)하는 낸드(NAND)형 플래쉬메모리에 관한 것이다.
- <13> 고집적 대용량에 유용한 낸드형 플래쉬메모리는 비휘발성메모리로서 현재 이동통신환경, 셋탑박스 또는 게임기 등에서 널리 사용되고 있으며, 그 응용범위가 증가하고 있다. 낸드형 플래쉬메모리에는 메모리로서의 기본적인 기능인 읽기, 쓰기(또는 프로그램) 및 소거 동작을 수행할 수 있으며, 새로운 응용에 부합하기 위하여 카피백(copy-back) 기능이 요구된다.
- <14> 카피백은 낸드형 플래쉬메모리에서 특정 어드레스의 페이지에 저장된 데이터를 다른 어드레스의 페이지로 옮기는 기능이다. 카피백 동작은, 특정 어드레스의 페이지에 저장된 데이터를 내부의 페이지버퍼에 임시로 저장한 다음 메모리의 외부로 독출하는 과정없이 저장된 페이지 데이터를 곧바로 쓰기(또는 프로그램) 데이터로 이용하여 다른 어드레스의 페이지에 저장한다. 카피백 기능을 이용함에 의해, 옮기고자 하는 데이터를 읽어내는 동작과 다시 특정 어드레스의 페이지에 대한 쓰기 동작을 위한 데이터 로딩동작을 생략할 수 있으므로, 시스템의 처리속도에 관련된 성능을 향상시킬 수 있다.

<15> 그러나, 도 1에 보인 바와 같이, 페이지 PG4에 저장된 데이터를 페이지 PGn-3으로 옮기는 카피백 과정에서, 읽고자 하는 페이지인 PG4와 쓰고자 하는 페이지인 PGn-3에 각각 1비트씩의 에러가 존재하는 경우가 있다. 그러한 상태에서, PG4의 데이터가 페이지버퍼 10으로 임시 저장된 후 PGn-3에 쓰여지게 되면, 결과적으로 2비트의 에러가 PGn-3에 나타나게 된다. 현재, 카드형 메모리컨트롤러의 대부분은 하나의 페이지당 1비트의 에러를 보정할 수 있도록 설계되는 것이 기본이지만, 도 1의 경우와 같이 카피백 후에 발생하는 2비트의 에러에 대하여는 복원하는 것이 불가능하다.

<16> 이와 같은 문제점을 해결하기 위하여, 카피백 기능이 내장된 플래쉬메모리를 채용하는 메모리 컨트롤러에 1페이지당 2비트의 에러를 정정할 수 있는 회로를 추가할 수는 있지만, 회로가 복잡하고 효율성이 떨어진다.

#### 【발명이 이루고자 하는 기술적 과제】

<17> 따라서, 본 발명의 목적은 낸드형 플래쉬메모리에서 카피백 동작에 따른 비트 에러를 줄일 수 있는 장치 및 방법을 제공함에 있다.

<18> 본 발명의 다른 목적은 낸드형 플래쉬메모리에서 카피백 과정에서 원래의 비트에러가 전사되는 것을 방지하는 장치 및 방법을 제공함에 있다.

<19> 전술한 본 발명의 목적을 달성하기 위하여, 본 발명에 실시예에 따른 낸드형 플래쉬메모리는, 페이지버퍼에 특정 페이지에 저장된 원데이터의 비트에러를 정정하는 회로, 상기 원데이터를 상기 회로로 제공하고 상기 회로에 의해 정정된 보정데이터를 상기 페이지버퍼로 제공하는 수단, 그리고 상기 원데이터를 상기 페이지버퍼로 복사하고 상기 페이지버퍼로부터 상기 보정데이터를 다른 페이지에 저장하는 수단을 포함한다.

- <20> 본 발명의 실시예에 따른 낸드형 플래쉬메모리는 상기 원데이타로부터 상기 보정데이타를 생성하고 전송하는 과정에서 별도의 버퍼를 사용하지 않고 낸드형 플래쉬메모리에서 통상의 프로그램 또는 독출동작을 위하여 사용되는 페이지버퍼를 이용한다.
- <21> 상기 원데이타는 프로그램과정에서 생성된 고유의 패리티를 갖고 있으며, 상기 회로는 상기 원데이타로부터 새로운 패리티를 생성하고 상기 새로운 패리티와 상기 고유의 패리티를 비교한다.
- <22> 또한, 본 발명의 실시예에 따른 카피백 방법은, 상기 원데이타를 상기 페이지버퍼에 저장하는 단계; 상기 페이지버퍼에 저장된 상기 원데이타로부터 새로운 패리티를 생성하는 단계; 상기 고유의 패리티와 상기 새로운 패리티를 비교하는 단계; 상기 비교결과에 응답하여 상기 원데이타에 대한 보정데이타를 형성하는 단계; 그리고 상기 보정데이타를 상기 페이지버퍼를 통하여 상기 다른 어드레스의 페이지로 옮기는 단계를 순차적으로 진행한다.

#### 【발명의 구성 및 작용】

- <23> 본 발명이 적용되는 메모리는 낸드형 플래쉬메모리로서, 집적회로 카드 등과 같은 휴대용 카드 장치에 채용될 수 있다.
- <24> 본 발명에 따른 낸드형 플래쉬메모리는 다수개의 메모리셀들이 스트링 형태로 배열되고, 메모리셀들의 스트링들은 행과 열로 배열된 워드라인들과 비트라인들에 연결되어 있다.
- <25> 본 발명에 따른 낸드형 플래쉬메모리는 또한 어드레스에 의해 지정되는 복수개의 페이지들로 구분된다. 각 페이지는 하나의 워드라인에 연결된 다수개의 메모리셀들로 구성되며, 하나의 페이지를 구성하는 메모리셀들의 비트라인들에 각각 연결된다. 각 페이지는 읽기 및 쓰기의 기본단위가 된다.





- <26> 본 발명에 따른 낸드형 플래쉬메모리는 메모리셀의 데이터를 지우는 소거동작, 일정한 페이지에 데이터를 쓰는 프로그램동작, 메모리셀에 저장된 데이터를 메모리의 외부로 읽어내는 독출동작, 소거 또는 프로그램된 데이터에 대한 검증동작, 그리고 메모리 외부로 데이터를 독출하지 않고 특정 페이지의 데이터를 다른 페이지로 옮겨 저장하는 카피백동작을 수행하도록 설계된다.
- <27> 이러한 기능적인 동작들을 수행하기 위하여, 본 발명에 따른 낸드형 플래쉬메모리는 프로그램, 독출, 또는 카피백 동작시에 데이터를 로딩하거나 메모리셀로부터 읽혀진 데이터를 일시적으로 저장하는 페이지버퍼회로를 가진다. 또한, 외부에서 제공되는 어드레스에 응답하여 페이지 또는 메모리셀의 지정하기 위한 버퍼 및 디코더와, 데이터값을 감지하기 위한 센스앰프, 데이터의 입출을 통제하는 게이트회로, 그리고 제동작들을 통제하고 관리하기 위한 제어회로를 포함한다.
- <28> 본 발명에 따른 실시예는 카피백 동작에 관련된 에러 감지 및 정정에 관하여 기술될 것이며, 낸드형 플래쉬메모리의 다른 동작들(소거, 프로그램, 독출 등)에 관하여는 생략하거나 간단히 설명할 것이다. 그러나, 본 발명에 따른 낸드형 플래쉬메모리에서 그러한 동작들이 당연히 수행될 수 있음을 이해하여야 한다.
- <29> 후술되는 본 발명의 실시예에서는, 외부로 데이터가 읽혀지는 동작을 "독출(read)"로 정의하고, 특정 페이지의 데이터가 페이지버퍼로 옮겨지는 동작을 "복사(copy)"로 정의하고, 그리고 페이지버퍼로부터 다른 페이지로 데이터가 옮겨지는 동작을 "전사"로 정의한다. 또한, 비트 에러를 가진 특정 페이지의 데이터를 "원데이터"로 정의하고, 에러정정되어 다른 페이지로 전사될 데이터를 "보정데이터"로 정의한다.

- <30> 이하, 본 발명에 따른 실시예를 첨부된 도면들 도 2 내지 도 9를 참조하여 설명한다.
- <31> 도 2는 본 발명에 따른 카피백시의 에러 감지 및 정정 과정을 개략적으로 보여준다. 먼저, 페이지 PG4에 저장된 데이터가 읽기 과정을 통하여 페이지버퍼 10에 저장된다. 페이지버퍼 10에 저장된 PG4의 데이터(즉, 원데이터)는 다른 페이지 PGn-3에 쓰이기(또는 전사되기) 전에 에러정정회로(ECC) 20를 통하여 에러정정된다. 그 후, 에러정정된 데이터(즉, 보정데이터)는 페이지버퍼 10을 통하여 다른 페이지 PGn-3에 프로그램된다. 그 결과, 원데이터가 보유하고 있던 에러 비트가 도 1의 경우와 같이 다른 페이지에 전사되는 현상이 방지된다.
- <32> 전사된 페이지 PGn-3에 자체적인 1비트의 에러가 있더라도 통상의 에러 감지 및 정정 기능에 의하여 보정될 것이다. 본 발명에 따른 카피백 동작에서의 에러 감지 및 정정 과정에 관한 구체적인 예는 후술될 것이다.
- <33> 도 3은 본 발명에 따른 카피백시의 에러 감지 및 정정을 위한 기능적인 구성과 상호연결 관계를 보여 주며, 도 4는 에러정정을 위하여 원데이터 및 보정데이터를 전송하기 위하여 페이지버퍼와 입출력라인(I/O) 사이에 배치된 래치/컬럼디코딩 블록(LDB)의 구성을 보여 준다.
- <34> 도 3을 참조하면, 페이지버퍼 10에 임시로 저장된 원데이터 De가 에러정정회로 20으로 제공되면, 이전의 프로그램과정에서 생성되어 일부 메모리영역에 저장되어 있던 구(old) 패리티 OP와 복사과정에서 새롭게 발생된 신(new) 패리티 NP를 비교기 50에서 비교한다. 여기서, 신 패리티 NP는 카피백동작 중에 발생하는 비트에러(진행성 비트에러)에 대한 정정을 위하여 발생하는 패리티 데이터이다. 구 패리티 OP와 신 패리티 NP는 동일한 방식으로 발생된다. 구 패리티 OP와 신 패리티 NP를 비교한 후, 에러가 있는 해당 어드레스에 관한 정보 Ae가 비교기 50으로부터 출력된다. 에러 어드레스정보 Ae를 입력한 에러정정 논리회로 60은 원데이터 De의

에러를 보정한 다음 보정데이터 Dc를 제어신호들 CNT와 함께 페이지버퍼 10으로 전송한다.

본 발명에 따른 패리티의 발생과 비교에 관하여는 후술될 것이다.

<35> 도 4는 페이지버퍼를 통한 원데이터 및 보정데이터의 입출력 경로를 보여 준다. 도 4를 참조하면, 특정 어드레스의 페이지의 원데이터는 각각의 입출력라인들 I/00~I/0n-1에 대응하여 배치된 래치/컬럼디코딩 블록들 LDB0~LDBn을 통하여 도 3의 에러정정회로 20으로 제공된다. 또한, 에러정정회로 20에 의한 보정데이터는, 래치/컬럼디코딩 블록들 LDB1~LDBn을 통하여 다른 페이지로 전사된다.

<36> 특정 어드레스의 페이지의 원데이터(De)는 페이지버퍼 센싱 블록 12에 의해 읽혀진 다음 각각의 비트라인들에 대응하는 래치들 LCH0~LCHn-1에 임시 저장된다. 래치들에 저장된 원데이터는 컬럼 디코딩에 의해 제어되는 게이트신호들 YAO~YAn-1 및 YB0~YBn-1에 응답하여 컬럼 게이트들(또는 Y-게이트들) AF0~AFn-1 및 BG0~BGn-1을 통하여 입출력라인들 I/00~I/0n-1으로 순차적으로 전송된다. 예컨대, 비트라인 BL0에 해당하는 원데이터 비트가 입출력라인 I/00를 통하여 에러정정회로 20로 전송되기 위해서는, 도 5에 보인 바와 같이, 게이트신호 YAO 및 YB0가 동시에 "하이"상태로 활성화되어 있어야 한다(Error Data Out).

<37> 에러정정회로 20의 동작에 의해 보정된 데이터(Dc)는 입출력라인들 I/00~I/0n-1을 통하여 입력되어 각각의 비트라인들에 대응하는 래치들에 일시저장된다. 예컨대, 비트라인 BL0에 대응하는 보정데이터 비트가 에러정정회로 20으로부터 입출력라인 I/00를 통하여 페이지버퍼로 전송되기 위해서는, 도 5에 보인 바와 같이, 게이트신호 YAO 및 YB0가 동시에 "하이"상태로 활성화되어 있어야 한다(Corrected Data In). 입출력라인들과 컬럼게이트들을 통하여 전송된 보정데이터는 상기 래치들에 저장된 다음 다른 페이지로 전사될 것이다.

<38> 도 6은 본 발명에 따라 패리티를 발생시키는 과정을 보여 준다. 패리티를 발생시키는 방식에는 병렬방식과 직렬방식이 있으나, 본 발명의 실시예에서는 회로적인 부담을 줄이기 위하여 직렬방식을 채용한다.

<39> 본 발명의 실시예에서 패리티발생의 대상이 되는 원데이터는 예컨대 8비트-512바이트로 구성된다. 본 발명의 실시예에서는, 하나의 바이트(Byte)를 구성하는 8비트에 대한 컬럼패리티와, 각각 8비트로 된 512바이트에 대한 라인패리티를 구한다.

<40> 컬럼패리티와 라인패리티를 발생하는 회로는 도 7과 도 8에 각각 도시되어 있으며, 이 회로들은 도 4의 패리티발생회로 40에 내장된다. 컬럼 또는 라인패리티의 발생은 해당하는 비트 또는 바이트에서의 비트에러를 검출하기 위하여 가능한 조합들을 익스클루시브-오아(Exclusive-OR; XOR) 논리 연산으로 처리한다.

<41> 먼저, 도 6 및 도 7과 아래의 표 1을 참조하여 컬럼패리티를 발생하는 과정을 설명한다. 표 1은 하나의 바이트를 구성하는 원데이터의 8비트에 대한 컬럼패리티 비트들 CP1, nCP1, CP2, nCP2, CP4, nCP4를 생성하기 위한 익스클루시브-오아 논리 조합들을 나타낸 것이다.

<42> 【표 1】

컬럼패리티	계 산 식							
b7 b6 b5 b4	b3 b2 b1 b0							
CP1	* *				* *			
nCP1	* *				* *			
CP2	* *				* *			
nCP2	* *				* *			
CP4	* * * *							
nCP4					* * * *			

<43> 표 1에 따른 6개의 컬럼패리티비트들의 연산식들은 다음과 같다. "\*"은 익스클루시브-오아 논리연산자를 나타낸다.

<44>  $CP1 = b7 * b5 * b3 * b1$

<45>  $nCP1 = b6 * b4 * b2 * b0$

<46>  $CP2 = b7 * b6 * b3 * b2$

<47>  $nCP2 = b5 * b4 * b1 * b0$

<48>  $CP4 = b7 * b6 * b5 * b4$

<49>  $nCP4 = b3 * b2 * b1 * b0$

<50> 이와같은 컬럼패리티 발생을 위한 연산조합을 구현하기 위하여, 도 7을 참조하면, 각 컬럼패리티비트는 4개의 XOR 게이트들(XR)과 하나의 플립플롭 FF를 통하여 발생된다. 각 입출력 라인은 각 데이터비트에 대응한다. 즉, 컬럼패리티비트 nCP4는, 입출력라인 I/00 및 I/01에 입력이 연결된 XR1, 입출력라인 I/02 및 I/03에 입력이 연결된 XR2, XR1 및 XR2의 출력에 입력이 연결된 XR13, XR13의 출력과 궤환되는 컬럼패리티비트 nCP4를 입력으로 하는 XR19, 그리고 XR19의 출력을 입력하여 컬럼패리티비트 nCP4를 발생하는 플립플롭 FF6을 통하여 발생된다. nCP4와 상보되는(complementary) 컬럼패리티비트 CP4는, 입출력라인 I/04 및 I/05에 입력이 연결된 XR3, 입출력라인 I/06 및 I/07에 입력이 연결된 XR4, XR3 및 XR4의 출력에 입력이 연결된 XR14, XR14의 출력과 궤환되는 컬럼패리티비트 CP4를 입력으로 하는 XR20, 그리고 XR20의 출력을 입력하여 컬럼패리티비트 CP4를 발생하는 플립플롭 FF5를 통하여 발생된다.

<51> 컬럼패리티비트 nCP2는, 입출력라인 I/00 및 I/01에 입력이 연결된 XR5, 입출력라인 I/04 및 I/05에 입력이 연결된 XR6, XR5 및 XR6의 출력에 입력이 연결된 XR15, XR15의 출력과 케환되는 컬럼패리티비트 nCP2를 입력으로 하는 XR21, 그리고 XR21의 출력을 입력하여 컬럼패리티비트 nCP2를 발생하는 플립플롭 FF4를 통하여 발생된다. nCP2와 상보되는 컬럼패리티비트 CP2는, 입출력라인 I/02 및 I/03에 입력이 연결된 XR7, 입출력라인 I/06 및 I/07에 입력이 연결된 XR8, XR7 및 XR8의 출력에 입력이 연결된 XR16, XR16의 출력과 케환되는 컬럼패리티비트 CP2를 입력으로 하는 XR22, 그리고 XR22의 출력을 입력하여 컬럼패리티비트 CP2를 발생하는 플립플롭 FF3를 통하여 발생된다.

<52> 컬럼패리티비트 nCP1은, 입출력라인 I/00 및 I/02에 입력이 연결된 XR9, 입출력라인 I/04 및 I/06에 입력이 연결된 XR10, XR9 및 XR10의 출력에 입력이 연결된 XR17, XR17의 출력과 케환되는 컬럼패리티비트 nCP1을 입력으로 하는 XR23, 그리고 XR23의 출력을 입력하여 컬럼패리티비트 nCP1을 발생하는 플립플롭 FF2를 통하여 발생된다. nCP1과 상보되는 컬럼패리티비트 CP1은, 입출력라인 I/01 및 I/03에 입력이 연결된 XR11, 입출력라인 I/05 및 I/07에 입력이 연결된 XR12, XR11 및 XR12의 출력에 입력이 연결된 XR18, XR18의 출력과 케환되는 컬럼패리티비트 CP1을 입력으로 하는 XR24, 그리고 XR24의 출력을 입력하여 컬럼패리티비트 CP1을 발생하는 플립플롭 FF1을 통하여 발생된다.

<53> 각 컬럼패리티비트가 해당하는 플립플롭의 전단에 위치한 XOR게이트로 케환입력되는 것은, 페이지버퍼에 복사된 원데이타에서 현재 데이타비트와 다음 데이타비트간의 변동을 검출(즉, 카피백 중에 발생하는 진행성 비트에러를 검출)하여 직렬로 처리하기 위함이다. 클럭신호 CLK와 리셋신호 RST가 플립플롭들 FF1~FF6에 공통으로 인가된다. 따라서, 플립플롭들 FF1~FF6은 클럭신호 CLK의 매 주기마다 상승에지에 응답하여 해당하는 컬럼패리티비트를 출력한다.

<54> 컬럼패리티를 발생함에 있어서, 가령 b3가 에러비트라고 가정하면, 컬럼패리티비트들 중 CP1, CP2 및 nCP4가 "1"로 나타날 것이다.

<55> 다음으로, 도 6 및 도 8과 아래의 표 2를 참조하여 라인패리티를 발생하는 과정을 설명한다. 표 2는 바이트 단위로 모두 512개의 바이트들에 대한 라인패리티 비트들 LP1, nLP1,..., LP512, nLP512를 생성하기 위한 이식배열-오아(XOR) 논리 조합들을 나타낸 것이다. XOR 논리연산을 위한 조합방식은 전술한 컬럼패리티들에 대한 방식과 동일함을 알 수 있다.

<56> 【표 2】

라인 패리티	계 산 식										
	B512	B511 B510 B509	..	B256	B255 B254	B253	..	B4	B3 B2	B1	
LP1	*	*	..	*	*		..	*	*		
nLP1		* *	..		*	*	..		*	*	
LP2	*	*	..	*	*		..	*	*		
nLP2		* *	..		*	*	..		*	*	
LP4	*	* * *	..	*	* *	*	..				
nLP4			..				..	*	* *	*	
..	...										
LP512	*	* * *	..				..				
nLP512			..	*	**	*	..	*	* *	*	

<57> 표 2에 따른 1024개의 라인패리티비트들의 연산식들은 다음과 같다. "\*"은 이식배열-오아 논리연산자를 나타내며, "^B"는 8개의 데이터비트들을 포괄하는 XOR연산결과를 나타내며(즉,  $\wedge B = b7 * b6 * b5 * b4 * b3 * b2 * b1 * b0$ ), 이를 이하 "단위 바이트 패리티비트"라 칭한다.

<58>  $LP1 = \wedge B512 * \wedge B510 * \dots * \wedge B256 * \wedge B254 * \dots * \wedge B4 * \wedge B2$

<59>  $nLP2 = \wedge B511 * \wedge B509 * \dots * \wedge B255 * \wedge B253 * \dots * \wedge B3 * \wedge B1$

<60>  $LP2 = \wedge B512 * \wedge B511 * \dots * \wedge B256 * \wedge B255 * \dots * \wedge B4 * \wedge B3$

<61>  $nLP2 = \wedge B510 * \wedge B509 * \dots * \wedge B254 * \wedge B253 * \dots * \wedge B2 * \wedge B1$

<62>  $LP4 = \wedge B512 * \wedge B511 * \wedge B510 * \wedge B509 * \dots * \wedge B256 * \wedge B255 * \wedge B254 * \wedge B253$

<63>  $\dots * \wedge B8 * \wedge B7 * \wedge B6 * \wedge B5$

<64>  $nLP4 = \wedge B508 * \wedge B507 * \wedge B506 * \wedge B505 * \dots * \wedge B252 * \wedge B252 * \wedge B250 * \wedge B249$

<65>  $\dots * \wedge B4 * \wedge B3 * \wedge B2 * \wedge B1$

<66> .....

<67>  $LP512 = \wedge B512 * \wedge B511 * \wedge B510 * \wedge B509 * \dots * \wedge B260 * \wedge B259 * \wedge B258 * \wedge B257$

<68>  $nLP512 = \wedge B256 * \wedge B255 * \wedge B254 * \wedge B253 * \dots * \wedge B4 * \wedge B3 * \wedge B2 * \wedge B1$

<69> 이와같은 라인패리티를 발생하기 위한 연산조합을 구현하기 위하여, 도 8을 참조하면, 한 바이트를 구성하는 8비트들에 대한 XOR연산결과를 XOR게이트들 XR31~XR37을 통하여 먼저 산출한다( $\wedge B$ ). 이로부터 512개의 라인패리티비트들 LP1~LP512 및 512개의 상보 라인패리티비트들 nLP1~nLP512를 발생하기 위하여 단위 바이트 패리티비트  $\wedge B$ 가 산출되는 XOR게이트 XR37의 출력을 1024개로 분지하고, 각각의 분지된 비트경로가 각각의 해당하는 클럭제어신호들 CLK1~CLK512 및 nCLK1~nCLK512에 응답하여 라인패리티비트들로 발생되도록 한다.

<70> 즉, XOR게이트 XR37로부터 발생된 단위 바이트 패리티비트  $\wedge B$ 는 낸드게이트들 ND1~ND1024에 공통으로 인가된다. 8개의 데이터비트들 중 어느 하나가 에러비트이면  $\wedge B$ 는 "1"



로 나타난다. 낸드게이트들 ND1~ND1024는 클럭제어신호들 nCLK1, CLK1, nCLK2, CLK2, ..., nCLK512, CLK512에 의해 그것들의 비트경로가 제어된다. 낸드게이트들 ND1~ND1024의 출력들은 XOR게이트들 XOR1~XOR1024에 각각 입력된다. XOR게이트들 XOR1~XOR1024는 또한 해당하는 라인패리티비트들 nLP1, LP1, ..., nLP512, LP512를 각각 입력한다. XOR1~XOR1024의 출력은 플립플롭들 FF1~FF1024을 통하여 해당하는 라인패리티비트들로 발생된다.

<71> 도 9의 타이밍도는 페이지버퍼 10으로부터 에러정정회로 20으로 원데이터를 보내기 위한 게이트제어신호들 및 클럭제어신호들의 상태를 보여준다. 도 9의 타이밍도는 전술한 도 4 내지 도 8의 회로동작에 관련되어 있다. 도 9의 타이밍도에서 진행되는 원데이터 전송과정은 도 4의 첫번째 래치/컬럼디코딩 블록 LDB1을 통하여 이루어지는 것을 일례로 한다.

<72> 먼저, 게이트제어신호 YB1이 하이레벨인 상태에서 게이트제어신호들 YA1~YAn이 순차적으로 하이레벨로 됨에 따라 각 비트라인에 대응하는 데이터비트들이 대응하는 입출력라인 I/O를 통하여 에러정정회로 20으로 전송된다. 마찬가지로, 게이트신호 YB0 내지 YBn-1이 순차적으로 각각 하이레벨인 동안 게이트신호들 YA0~YAn-1이 순차적으로 하이레벨로 되면서 n개의 비트들이 입출력라인을 통하여 에러정정회로로 공급되어, 총 512 바이트의 원데이터가 에러정정회로로 제공된다.

<73> 한편, 에러정정회로 20에서는, 클럭신호 CLK가 일정한 주기로 발진(oscillating)함에 따라 도 7에 보인 컬럼패리티 발생회로의 플립플롭들 FF1~FF6은 컬럼패리티비트 CP1/nCP1~CP4/nCP4들을 출력한다. 클럭신호 CLK로부터 분주되어 생성되는 클럭제어신호들 CLK1/nCLK1~CLK512/nCLK512에 응답하여 도 8에 보인 라인패리티제어회로의 낸드게이트들 ND1~ND1024가 비트경로를 활성화시키고, 클럭신호 CLK에 응답하여 플립플롭들 FF1~FF1024로부터 라인패리티비트들 LP1/nLP1~LP512/nLP512가 발생된다.

- <74> 도 10은 본 발명에 따른 카피백중의 에러정정 과정을 일괄적으로 보여준다. 먼저, 통상적인 프로그램과정에서 패리티(구 패리티)를 발생하여(단계 S1) 이를 메모리의 스페어영역에 저장한다(단계 S2). 그 다음, 특정 어드레스의 페이지로부터 원데이터를 페이지버퍼로 복사하는 동안 전술한 과정(도 6 내지 도 8)을 통하여 신패리티를 발생시킨 후(단계 S3), 에러정정회로 20의 비교기 50에서 원 패리티 OP와 신 패리티 NP를 비교한다(단계 S4). 512 바이트의 데이터에 대하여 구 패리티 OP와 신 패리티는 예를 들면 24비트로 동일한 비트수이다.
- <75> 구 패리티와 신 패리티를 비교한 결과, 구 패리티 및 신 패리티의 각 24비트가 모두 동일하면(즉, XOR연산에 의한 비교결과값이 "0"이면) 에러가 없는 것으로 처리된다(단계 S5). 그러나, 24비트 중에 1/2인 12비트에 대한 비교결과값이 "1"이면 1-비트에러로 감지된다. 한편, 24비트 중에 1비트에 대한 비교결과값만이 "1"로 나타나면, 이는 원래의 1-비트에러로서 복사된 페이지에서 원래 갖고 있던 치유 가능한 단일에러로 인식된다. 다른 한편으로, 전술한 다른 상태로 비교결과가 나타나는 경우에는 적어도 2비트 이상의 에러가 있는 경우이다.
- <76> 이와 같은 에러상태에 대하여 커맨드에 응답하게 하여 사용자로 하여금 알 수 있도록 하는 것이 가능하다. 또한, 본 발명의 실시예에서 보인 에러정정회로 20은 본 발명에 따른 낸드형 플래쉬메모리에 내장되어 있다.
- <77> 전술한 실시예에서 보인 본 발명의 수단 또는 방법에 준하여 본 발명의 기술분야에서 통상의 지식을 가진 자는 본 발명의 범위내에서 본 발명의 변형 및 응용이 가능하다.

【발명의 효과】

- <78> 본 발명의 실시예에 의하면, 특정어드레스의 페이지에 저장된 원데이터에 존재하는 에러 비트를 정정한 다음 다른 페이지로 프로그램함에 의해, 카피백 후에 원데이터의 에러비트가 전사되는 것을 방지하는 효과가 있다.
- <79> 또한, 본 발명의 실시예에 의하면, 에러정정을 위한 별도의 버퍼를 사용하지 않고 낸드형 플래쉬메모리에서 사용되는 페이지버퍼를 이용하기 때문에, 카피백시의 에러정정을 위한 회로 구성상의 부담을 줄이는 이점이 있다.
- <80> 또한, 본 발명의 실시예에 의하면, 카피백 동작에서 뿐만 아니라 정상동작시에 페이지버퍼를 통한 데이터의 이동과정에서 발생될 수 있는 진행성 비트에러를 치유할 수 있는 수단을 제공하는 효과가 있다.

【특허청구범위】

【청구항 1】

비휘발성메모리에 있어서:

데이타를 저장하는 복수개의 페이지들;

페이지단위의 데이타를 임시로 저장하는 페이지 버퍼;

상기 페이지버퍼에 특정 페이지에 저장된 원데이타의 비트에러를 정정하는 회로;

상기 원데이타를 상기 회로로 제공하고 상기 회로에 의해 정정된 보정데이타를 상기 페이지버퍼로 제공하는 수단; 그리고

상기 원데이타를 상기 페이지버퍼로 복사하고 상기 페이지버퍼로부터 상기 보정데이타를 다른 페이지에 저장하는 수단을 구비함을 특징으로 하는 비휘발성메모리.

【청구항 2】

제1항에 있어서,

상기 원데이타가 고유의 패리티를 갖고 있음을 특징으로 하는 비휘발성메모리.

【청구항 3】

제2항에 있어서,

상기 회로가, 상기 원데이타로부터 새로운 패리티를 생성하고, 상기 새로운 패리티와 상기 고유의 패리티를 비교함을 특징으로 하는 비휘발성메모리.

【청구항 4】

제3항에 있어서,

상기 회로가:

상기 원데이터에서 하나의 바이트를 구성하는 비트들에 대한 컬럼패리티를 생성하는 회로; 그리고

상기 원데이터에서 소정갯수의 비트들로 구성된 바이트들에 대한 라인패리티를 생성하는 회로를 구비함을 특징으로 하는 비휘발성메모리.

【청구항 5】

제1항에 있어서,

상기 비휘발성메모리가 낸드형 플래쉬메모리임을 특징으로 하는 비휘발성메모리.

【청구항 6】

비휘발성메모리에 있어서:

데이터를 저장하는 복수개의 페이지들로 구성된 데이터 저장영역;

상기 페이지들에 대한 프로그램 동작 중에 발생된 제1패리티를 상기 데이터 저장영역의 스페어영역에 저장하는 제1수단;

페이지단위의 데이터를 저장하는 페이지버퍼;

상기 페이지들 중 특정 어드레스의 페이지에 저장된 원데이터를 상기 페이지버퍼에 복사하는 제2수단;

상기 페이지버퍼에 저장된 상기 원데이터로부터 제2패리티를 생성하는 제3수단; 그리고

상기 제1패리티와 상기 제2패리티를 비교한 결과에 응답하여 상기 원데이터의 보정데이터를 상기 페이지버퍼로 전송하는 제4수단을 구비함을 특징으로 하는 비휘발성메모리.

【청구항 7】

제6항에 있어서,

상기 페이지버퍼에 저장된 보정데이터를 상기 페이지들 중 다른 어드레스의 페이지에 저장하는 제5수단을 더 구비함을 특징으로 하는 비휘발성메모리.

【청구항 8】

제6항에 있어서,

상기 제2패리티가 컬럼패리티와 라인패리티로 이루어짐을 특징으로 하는 비휘발성메모리.

【청구항 9】

제8항에 있어서,

상기 제3수단이:

상기 원데이터에서 하나의 바이트를 구성하는 비트들에 대한 컬럼패리티를 생성하는 회로; 그리고

상기 원데이터에서 소정갯수의 비트들로 구성된 바이트들에 대한 라인패리티를 생성하는 회로를 구비함을 특징으로 하는 비휘발성메모리.

【청구항 10】

제6항에 있어서,

상기 비휘발성메모리가 낸드형 플래쉬메모리임을 특징으로 하는 비휘발성메모리.

【청구항 11】

페이지단위의 데이터를 저장하는 페이지버퍼를 가지는 비휘발성메모리에서 특정 어드레스의 페이지에 저장되며 고유의 패리티를 포함하는 원데이터를 다른 어드레스의 페이지로 옮기는 방법에 있어서:

상기 원데이터를 상기 페이지버퍼에 저장하는 단계;

상기 페이지버퍼에 저장된 상기 원데이터로부터 새로운 패리티를 생성하는 단계;

상기 고유의 패리티와 상기 새로운 패리티를 비교하는 단계;

상기 비교결과에 응답하여 상기 원데이터에 대한 보정데이터를 형성하는 단계; 그리고

상기 보정데이터를 상기 페이지버퍼를 통하여 상기 다른 어드레스의 페이지로 옮기는 단계가 순차적으로 이루어짐을 특징으로 하는 방법.

【청구항 12】

제11항에 있어서,

상기 원데이터를 상기 페이지버퍼에 저장하기 전에, 상기 원데이터에 대한 상기 고유의 패리티를 상기 메모리의 스페어영역에 저장하는 단계를 더 구비함을 특징으로 하는 방법.

【청구항 13】

제11항에 있어서,

상기 비교결과에 따른 상태를 외부에서 알 수 있도록 하는 단계를 더 구비함을 특징으로 하는 방법.

【청구항 14】

비휘발성메모리에 있어서:

데이타를 저장하는 복수개의 페이지들로 구성된 데이타 저장영역;

특정 페이지의 데이타를 저장하며 상기 데이타 저장영역과 연결되는 페이지버퍼;

상기 페이지버퍼와 연결되며 상기 데이타의 비트불량을 감지하는 비트불량 감지수단과  
상기 비트불량을 보정하는 비트불량 보정수단을 포함하는 에러정정수단을 구비함을 특징으로  
하는 비휘발성메모리.

【청구항 15】

제14항에 있어서,

상기 비트불량 감지수단은, 상기 페이지버퍼에 저장된 데이타를 입력으로 하여 새로운  
패리티를 생성하는 패리티발생회로와, 상기 새로운 패리티와 상기 데이타에 포함되어 있는 고  
유의 패리티를 비교하여 에러 어드레스정보를 발생하는 비교기를 포함함을 특징으로 하는 비휘  
발성메모리.

【청구항 16】

제15항에 있어서,

상기 에러 어드레스정보는, 상기 비트불량 보정수단으로 입력되어 상기 데이타를 보정하  
고 보정된 데이타를 상기 페이지버퍼로 전송함을 특징으로 하는 비휘발성메모리.

【청구항 17】

제16항에 있어서,

상기 보정된 데이타의 전송은 제어신호에 의하여 제어됨을 특징으로 하는 비휘발성메모  
리.



【청구항 18】

제17항에 있어서,

상기 보정된 데이터는 상기 특정 페이지와 다른 페이지에 복사됨을 특징으로 하는 비휘발성메모리.

【청구항 19】

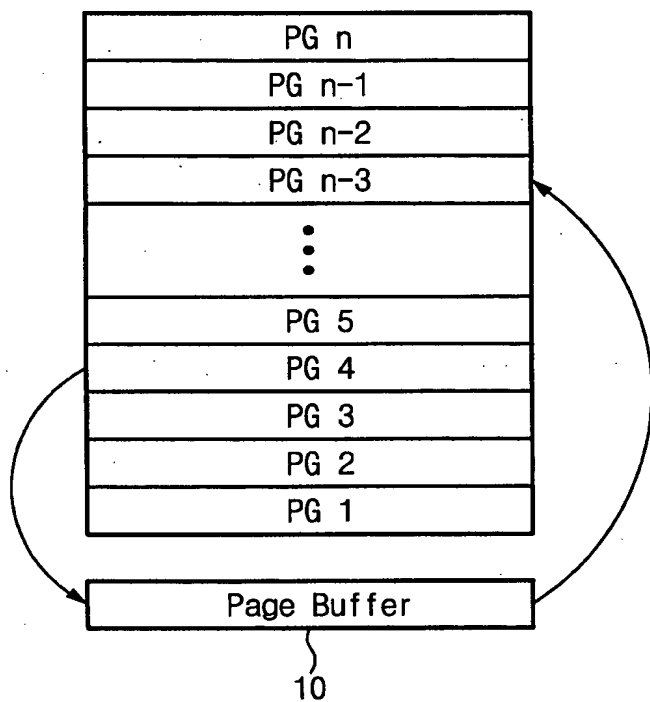
제14항에 있어서,

상기 비휘발성메모리가 낸드형 플래쉬메모리임을 특징으로 하는 비휘발성메모리.

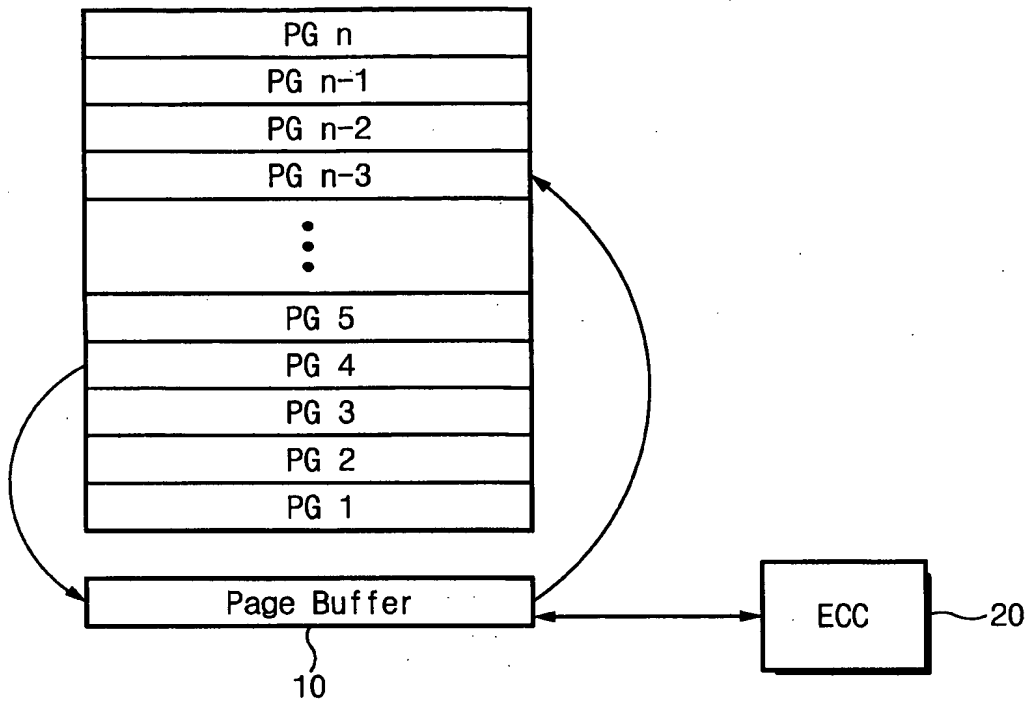
【도면】

【도 1】

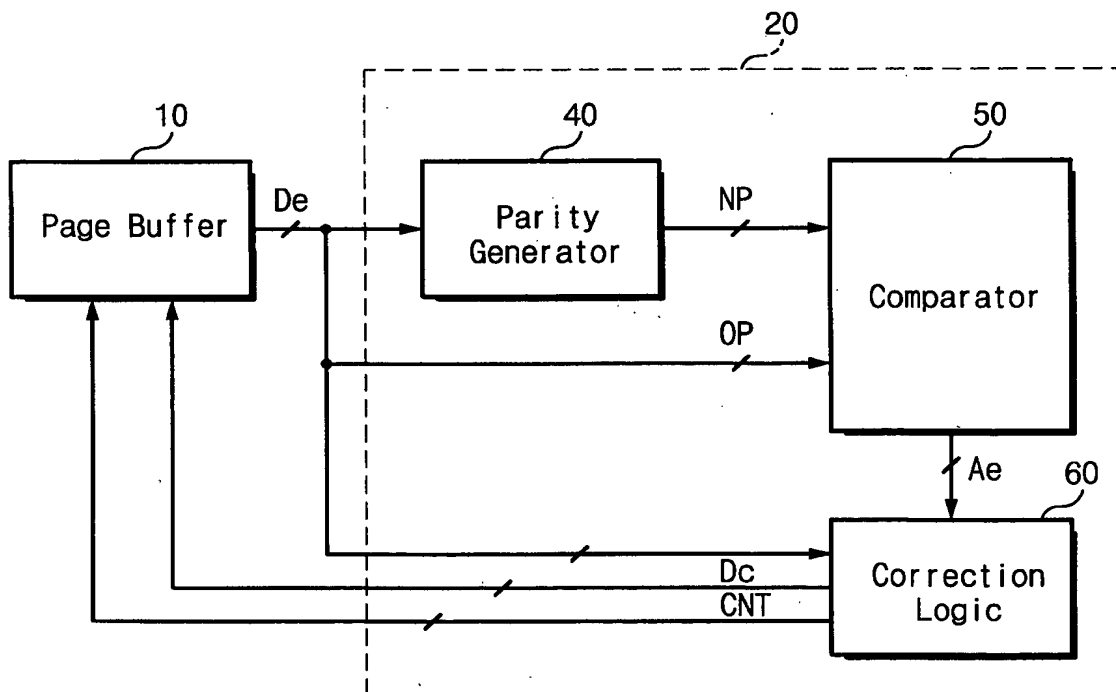
(종 래 기 술)



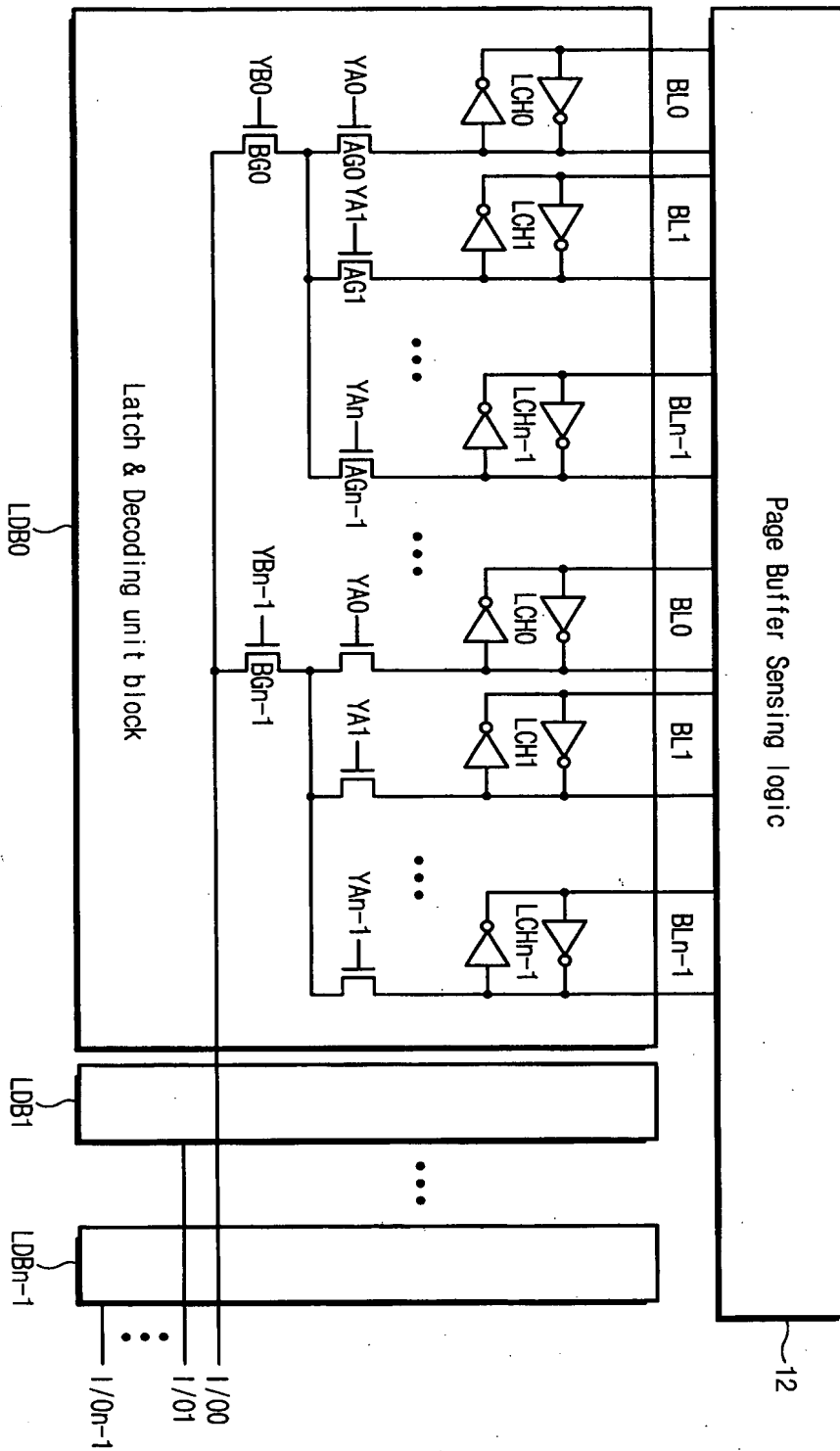
【도 2】



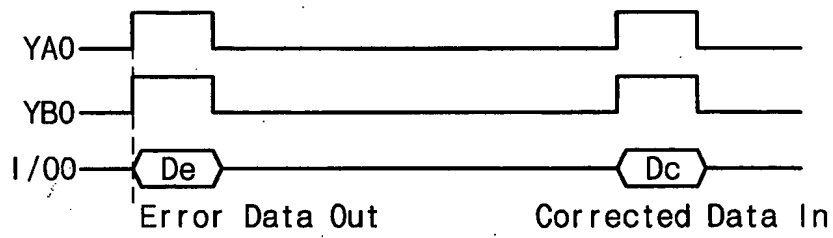
【도 3】



【도 4】



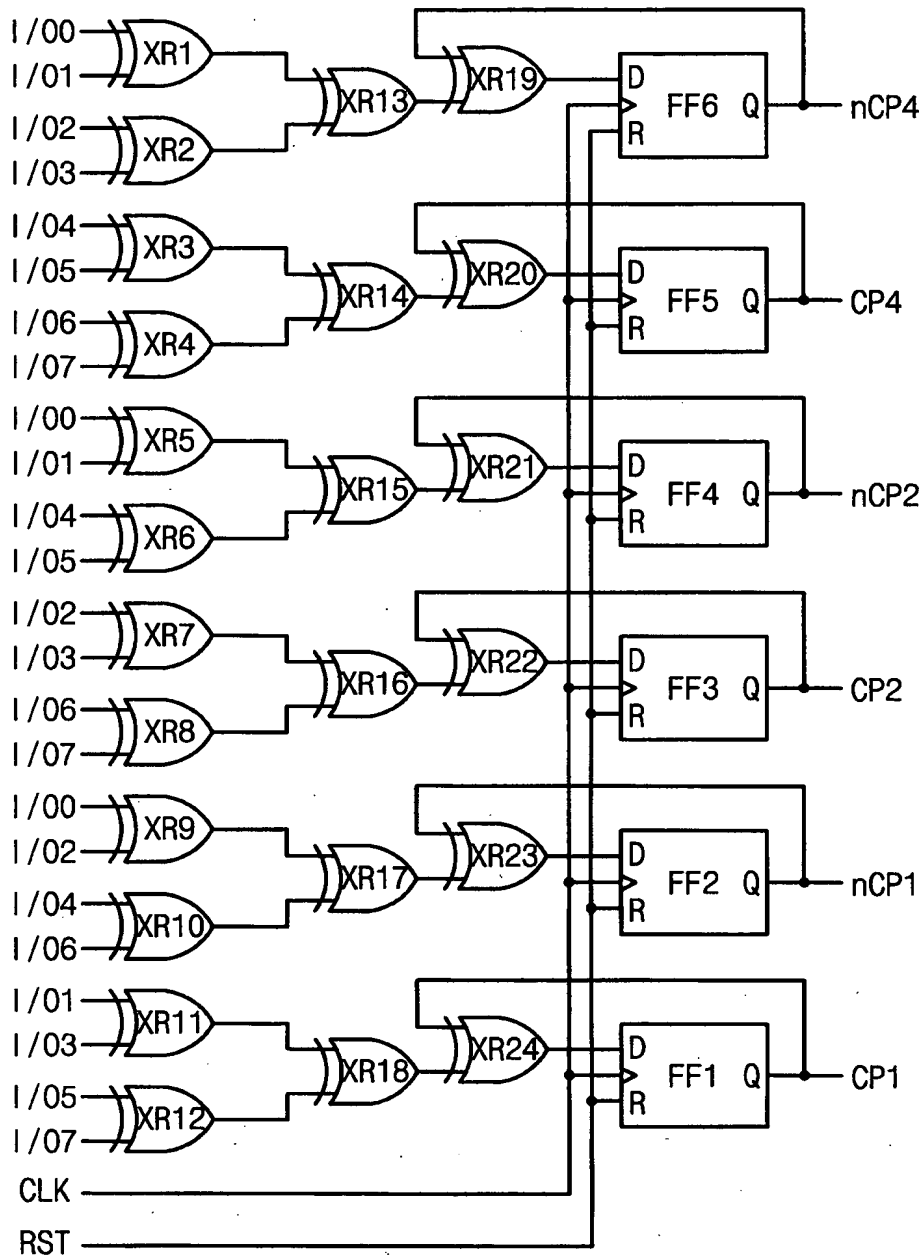
【도 5】



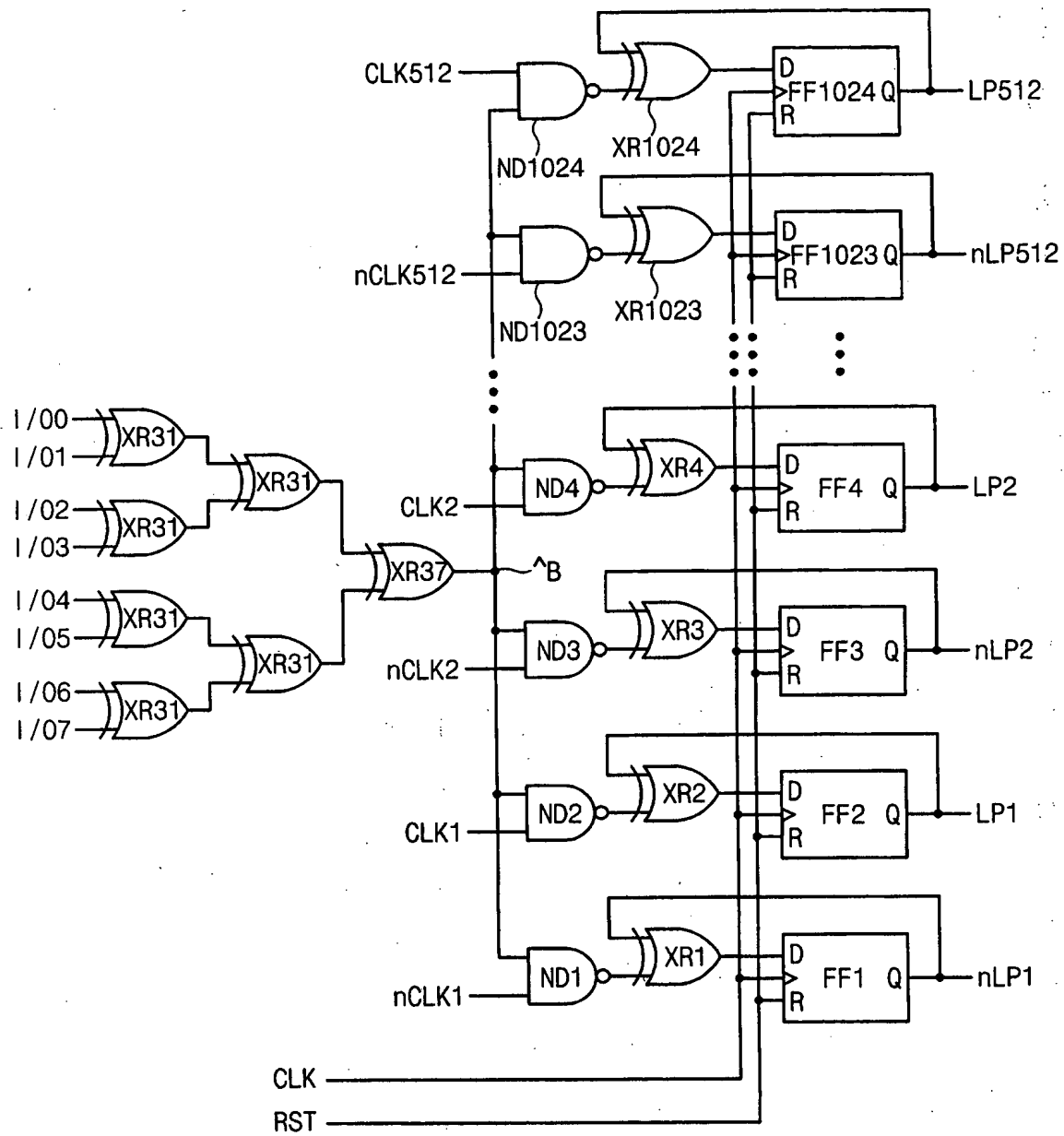
【표 6】

B1	b7	b6	b5	b4	b3	b2	b1	b0	nLP1	nLP2	nLP4	nLP512
B2	b7	b6	b5	b4	b3	b2	b1	b0	LP1			
B3	b7	b6	b5	b4	b3	b2	b1	b0	nLP1	LP2		
B4	b7	b6	b5	b4	b3	b2	b1	b0	LP1	LP2		
	∴	∴	∴	∴	∴	∴	∴	∴	∴	∴	∴	
B509	b7	b6	b5	b4	b3	b2	b1	b0	nLP1	nLP2		
B510	b7	b6	b5	b4	b3	b2	b1	b0	LP1		LP4	
B511	b7	b6	b5	b4	b3	b2	b1	b0	nLP1			
B512	b7	b6	b5	b4	b3	b2	b1	b0	LP1	LP2		LP512
CP1	nCP1	CP1	nCP1	CP1	nCP1	CP1	nCP1					
CP2	nCP2		CP2		nCP2							
CP4					nCP4							

【도 7】



【도 8】





【图 9】

